9 Graphs

9.1 Graphs and Graph Models

1. a graph $G = (V(G), E(G))$ consists of a set $V(G)$ of vertices, and a set $E(G)$ of edges (edges are pairs of elements of $V(G)$)

2. an edge is present, say $e = \{u, v\}$ (or simply just uv) for $u, v \in V(G)$, if there is a particular property between the two vertices (as we did for relations in Chapter 8)

3. an infinite graph is a graph with an infinite vertex set. Otherwise it is finite

4. a simple graph (or just a graph) is a graph such that each edge connects two different vertices, no two edges connecting the same pair (i.e. no parallel edges and also no loops)

5. multigraphs are graphs with parallel edges (multiple edges between the same two vertices)

6. pseudographs are graphs that contain loops and possibly parallel edges

7. a digraph (or directed graph) $D = (V(D), E(D))$ consists of a set $V(D)$ of vertices, and a set $E(D)$ of oriented arcs (arcs are ordered pairs of elements of $V(D)$, so $(u, v) \neq (v, u)$)

8. an oriented graph is a simple graph whose edges are oriented (i.e. no oriented loops, no double arcs)

9. a directed multigraph is a directed graph that may have multiple arcs and/or loops

10. structure of the graph is given by

    - edges: oriented or not
    - multiplicity of edges/arcs
    - loops: present or not

11. graph models:

    - friendship graph (or acquaintanceship graph): $V(G)$ is a set of people, and an edge is present if the two people are friends/know each other (These "social graphs" are associated with the phrase "six degrees of separation": the acquaintanceship graph connecting the entire human population has a diameter of six or less, i.e. any two people are connected by a chain of at most 6 friendships, i.e. 5 intermediate vertices that connect the two people).

- collaboration graph (centered on Paul Erdos, who was the most prolific mathematician ever): $V(G)$ are mathematicians, and an edge between two vertices is present if the two people wrote an article together. For any vertex of the graph, the minimum number of edges traveled from Erdos's node to the particular node is termed the Erdos number. Erdos himself has an Erdos number of 0. All those who co-authored a paper with him have Erdos number 1, and so on. The graph is not connected, as some people work alone. There is a similar graph for graph theorists (centered at Frank Harary).
- Hollywood graph (centered at Kevin Bacon): $V(G)$ is the set of actors, and two vertices are adjacent if the corresponding actors worked together on a movie. There is a similar notion of Bacon's number. This graph is connected, and it has about half a million vertices, representing actors (most actors can reach Bacon's vertex on average in 3 steps).
- There is now an Erdos-Bacon graph as well, with the sum of the values that it takes a person to get to Erdos and to Bacon (smallest number is 3)

12. digraphs models

    - Round Robin Tournament digraphs: they represent tournaments in which each team plays each of the other teams. Then $V(D)$ is the set of teams, and an arc (x, y) means that team x beat team y. This is a (simple) digraph, since there are no loops (no team plays against itself), and there are no multiple arcs (each pair of teams plays only one game)
    - call digraphs: $V(D)$ is the set of phone numbers, and an arc (x, y) represents a phone call where x called y. Note that if the calls were long distance calls, then orientation matters, because x gets charged for it (for local or cell phone calls orientation might not be needed). This is a directed multigraph since more than one call can be placed between a pair of numbers x and y. However, there are no loops.
    - web digraph: $V(D)$ is the set of webpages, and a link between two pages is represented by an arc (x, y) (if we care that page x has a link to page y), or by an edge $\{x, y\}$ (if all that matters is that the two pages are linked). This could have multiple arcs, and possibly loops.
    - road maps: $V(D)$ is the set of intersections of roads, and the arcs between two vertices represent streets going in that direction. This digraph is a multigraph, with possible loops.
    - most of the graphs we work with are simple graphs, and so we refer to them just as graphs (versus simple graphs).

9.2 Graph Terminology and Special Types of Graphs

For the rest of the chapter, we use G to denote a graph, $V(G)$ for its vertex set, $E(G)$ for the edge set, the lower case letter u, v, x, y, z to denote vertices in the graph, and the lower cases e, f to denote edges of the graph. For digraphs, we normally use D with $V(D)$ and $E(D)$ respectively. Vertices and arcs of the digraph use the same letters as in the graphs

1. u and v are adjacent if $\{u, v\} \in E(G)$ (or $uv \in E(G)$), and we say that u (or v) is incident with the edge $e = uv$ (or $e = \{u, v\}$) and e connects u and v

2. the degree of the vertex v, $\deg v$, is the number of edges incident with v

3. if $\deg v = 0$ then v is an isolated vertex

4. if $\deg v = 1$ then v is a pendant or an end vertex

5. The Handshaking Theorem (also called The First Theorem of Graph Theory): For a simple graph $G = (V(G), E(G))$, we have that

$$\sum_{v \in V(G)} \deg v = 2|E(G)|,$$

   where $|E(G)|$ is the cardinality of the edge set.

6. in particular, the above theorem says that the sum of the degrees of the vertices in a graph has to be even

7. Thm: A simple graph has an even number of vertices of odd degree (this is true since otherwise the sum of the degrees of the vertices would be odd, as the odd degree vertices are the only ones that can make the sum odd)

8. in a digraph D, the vertex v is adjacent to u if we have the arc (v, u) in D. Also, the vertex u is adjacent from v (v is the initial vertex of the arc, and u is the terminal vertex)

9. in a digraph D, the in-degree of v, $\deg^-(v)$ is the number of arcs that have v as a terminal vertex (i.e. the number of arcs coming into v). Also, the out-degree of v, $\deg^+(v)$ is the number of arcs that have v as an initial vertex (i.e. the number of arcs coming out of v)

10. Thm: For a digraph $D = (V(D), E(D))$, we have that

$$\sum_{v \in V(D)} \deg^- v = \sum_{v \in V(D)} \deg^+ v = |E(D)|.$$

    Note that this is true since each arc has an initial vertex and a final vertex, and so each arc contributes exactly 1 to each of the sums above.

11. the underlying graph G of a digraph D, is the graph obtained from D by removing the orientation of the arcs (this graph could be simple or a multigraph)

12. classes of **simple** graphs: they are SIMPLE graphs that have a particular property (each class is infinite)

    - complete graph, $K_n, n \geq 1$ is the graph on n vertices that has every possible edge present
    - path, $P_n, n \geq 2$ is the graph on n vertices $v_1, v_2, \ldots, v_n$ such that $v_i v_{i+1} \in E(G)$ for $1 \leq i \leq n-1$ (i.e. it has the consecutive edges $v_1v_2, v_2v_3, v_3v_4, \ldots, v_{n-1}v_n$)
    - cycle, $C_n, n \geq 3$ is the graph on n vertices $v_1, v_2, \ldots, v_n$ such that $v_i v_{i+1} \in E(G)$ for $1 \leq i \leq n$ where addition is performed modulo n (i.e. it has the consecutive edges $v_1v_2, v_2v_3, v_3v_4, \ldots, v_{n-1}v_n, v_nv_1$)
    - wheel, $W_n, n \geq 3$ (or $W_{1,n}$) is the graph obtained from the cycle C_n and a vertex v, by adding an edge between v and each vertex of the cycle.
    - n-Cube $Q_n, n \geq 1$ is that graph that has vertices represent all the binary n-strings, and two edges are adjacent if the two corresponding binary strings differ in just one bit (i.e the Hamming distance between the two bits is 1).
    - complete bipartite graph, $K_{a,b}, a, b \geq 1$ is the graph obtained by partitioning the vertices into two subsets of cardinality a and b ($n = a + b$) and all edges between any vertex of the first partite set and the second partite set are present (so the number of edges is ab). If either a or b is 1, we call the bipartite graph a star

13. bipartite graph: is a graph whose vertex set is partitioned into two subsets V_1 and V_2 ($V(G) = V_1 \cup V_2$), such that edges of the graph go between a vertex in V_1 and a vertex in V_2 (note that not every vertex of V_1 is adjacent to each vertex of V_2, unless we have a complete bipartite graph)

14. Thm: G is bipartite $\iff$ $V(G)$ can be colored with exactly two colors, where no two adjacent vertices are colored the same.

15. Thm: G is bipartite $\iff$ G contains no odd cycle.

16. Bipartite graphs are useful in modeling assignment of jobs to employees using a matching (where an edge xy is present if the employee x has the necessary training and knowledge to do job y), local networks, and others

17. a matching is a collection of edges (set of edges) so that no two edges share a common vertex. A maximum matching is a matching of maximum cardinality

18. a subgraph $H = (V(H), E(H))$ of a graph $G = (V(G), E(G))$ is a graph such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ (note that only edges whose end vertices are in $V(H)$ can be present in $E(H)$)

19. a proper subgraph $H = (V(H), E(H))$ of a graph $G = (V(G), E(G))$ is a subgraph that is not the whole graph (i.e is a graph such that $E(H) \subset E(G)$)

20. a supergraph $K = (V(K), E(K))$ of a graph $G = (V(G), E(G))$ is a graph such that $V(G) \subseteq V(K)$ and $E(G) \subseteq E(K)$ (note that only edges whose end vertices are in $V(K)$ can be present in $E(K)$). A supergraph is proper if it is different than the original graph G.

21. the union, $G \cup H$, of two graphs G and H is the graph whose vertex set is the set $V(G) \cup V(H)$, and the edge set is $E(G) \cup E(H)$. Note that $G \cup H$ in not connected (it has more pieces/components). You can define the intersection graph and difference graph similarly.

9.3 Graph Isomorphism

1. helps identify graphs that have the same "shape" but the way they are drew or referred to may be different (for example the cycle C_4 and the complete bipartite graph $K_{2,2}$)

2. representing graphs:

   - using a drawing along with a description (or just the class it belongs to)
   - using adjacency list: list the vertices, and then the edges that are incident with each vertex
   - using adjacency matrix: a matrix obtained by listing all the vertices along the top and the side of the matrix, and an entry a_{ij} of the matrix is 1 if and only if the two vertices v_i and v_j are adjacent. The matrix is unique up to the labeling used on the vertices (so there are $n!$ different adjacency matrices for a graph with n vertices). The adjacency matrix of a simple graph is symmetric since the adjacency relation is symmetric. However, for a digraph, the adjacency matrix does not have to be symmetric. (if the graph is simple, then the matrix is a $0 - 1$ matrix)
   - using incidence matrix: a matrix that shows what edges are incident to what vertices. It is an $n \times m$ matrix, where n is the number of vertices, and m is the number of edges/arcs

3. now the main question of the section is: How do we know if what may look like two different graphs is actually the same graph, just presented with a different drawing or a formula. This uses the concept of isomorphism

4. two simple graphs G and H are isomorphic, if there is a one-to-one correspondence f (a bijection) from $V(G)$ to $V(H)$ that preserves adjacencies (i.e. u and v are adjacent vertices in G if and only if $f(u)$ and $f(v)$ are adjacent in H). This function f is called an isomorphism from G to H.

5. How do we know if two graphs are isomorphic? One needs to find a function f as described above. To show the graphs are non isomorphic might be easier: see if there is the same degree sequence in the two graphs (or the same number of edges and vertices), or show that one graph has a property that the other doesn't (for example one graph might be bipartite and the other might not)

6. a graph invariant is a property that is preserved by an isomorphism, such as the number of vertices, number of edges, degree of a vertex, same number of vertices of a particular degree, same number of cycles, whether the graphs are one piece or not (i.e. connected–see next section)...

9.4 Connectivity

1. we determine whether a message can be sent between two computers using only intermediate links (sending a message along a path)

2. a $u - v$ path or walk in a graph is a sequence of edges that begin at vertex u and end at vertex v

3. a simple $u - v$ path/walk in a graph is a path that does not repeat edges either

4. a circuit or a cycle is a path that begins and ends with the same vertex (some of the vertices and edges may be repeated). If no edges are repeated then we have a simple cycle/circuit

5. same terminology holds for directed graphs, where we're allowed to travel in the direction that the arc points (like one way streets). In directed graphs it usually helps if the edges are listed using the vertices notation (for example the arc $(u_i v_i)$ instead of using the arc e)

6. a graph is connected if there is a path (or simple path) between any two vertices of the graph (the k^{th} power of the adjacency matrix will give the paths of length k)

7. a connected component of a graph G is a connected subgraph of G that is not a proper subgraph of any other connected subgraph of G (i.e. connected components are the largest connected subgraphs of G)

8. some vertices have important role in a graph, and the removal of a particular vertex in a network could cause lots of damage to its connectivity and thus to its usefulness. An example of such vertex is the cut vertex. A cut vertex in a connected graph G is a vertex whose removal produces more components.

9. similarly, a bridge is an edge of G whose removal produces more components.

10. an edge is a bridge iff it belongs to no simple cycle.

11. a digraph is strongly connected if for any two vertices of D, say u and v, there is a $u - v$ directed path and a $v - u$ directed path

12. if there is a directed simple circuit that contains all the vertices of D then D is strongly connected

13. a digraph is weakly connected if its underlying graph is connected (the underlying graph is the graph/multigraph obtained after removing the orientation of the arcs)

14. the strongly connected components of a digraph are the largest sub-digraphs that are strongly connected.

15. as mentioned before, paths and cycles can help determine whether two given graphs are isomorphic or not, by determining if they have the same number of paths/cycles of same length (the length is the number of edges in a path/cycle)

16. in counting the number of paths of length r between two vertices v_i and v_j, one can use the rth power of the graph's adjacency matrix: the a_{ij} entry in A^r tells the number of paths of length r between v_i and v_j (proof by induction on r)

9.5 Euler and Hamiltonian Paths

1. Given a graph, is it possible to travel along each edge and return to the start point without repeating an edge? Also, is it possible to travel through all the vertices (you don't have to use every edge) without repeating a vertex? Both of these questions arose in puzzles, but there are also practical applications such as traveling salesman problem or the garbage pickup routes

2. an Eulerian circuit in a graph G is a simple circuit containing every edge of G (note that vertices may be repeated as long as no edge is repeated)

3. an Eulerian path is a simple path (walk) that contains every edge of G (it ends at a different point than the starting point, and it may repeat vertices, but it may not repeat edges)

4. a Hamiltonian circuit is a simple circuit that travels through every vertex of the graph, without repeating a vertex (or edge).

5. a Hamiltonian path is a simple path that covers every vertex of the graph, ending point being different than the starting one (may not repeat edges nor vertices)

6. There are necessary and sufficient conditions for Eulerian paths and circuits, but not for the Hamiltonian ones

7. Characterization for Eulerian circuits A connected multigraph has an Eulerian circuit iff the degree of every vertex is even.

8. Characterization for Eulerian path: A connected multigraph has an Eulerian path iff there are exactly two vertices of odd degree.

9. Dirac's Thm Let G be a simple graph with n vertices.

   $$\text{If } \deg v \geq \lceil \tfrac{n}{2} \rceil, \forall v \in V(G), \text{ then } G \text{ has a Hamiltonian circuit.}$$

10. Ore's Thm Let G be a simple graph with n vertices.

    $$\text{If } \deg u + \deg v \geq n, \forall v, u \in V(G) \text{ such that } uv \notin E(G), \\ \text{then } G \text{ has a Hamiltonian circuit.}$$

11. a graph that has an Eulerian/Hamiltonian circuit is also called *Eulerian/Hamiltonian graph*

12. the contrapositive of the Dirac's and Ore's Theorems can help in proving that there is no Hamiltonian circuit

9.6 Shortest Path Problems

1. this section considers weighted graphs (graph whose edges have weights), that model a situation where the distance/time/money that it takes to get from one place to another matters. The distance/time/money will give the weights of the edges of the graph. This concept is useful in computer networks (the weights could be time that it takes a computer to respond or do a task), or for airline companies (where the weights could display the time or fuel, or both, that it takes to fly from point A to point B)

2. the most common problem in weighted graphs is finding a path of shortest length between two vertices, where the length of the path is the sum of the weights of the edges traveled (in an unweighted graph, the length of a path is the number of edges traveled).

3. Dijkestra's Algorithm is an algorithm that finds a shortest path from vertex v to vertex u, by finding vertex v_1 that is closest to v (let $S_1 = \{v, v_1\}$), then finding v_2 that is 2nd closest to v ($v_2 \neq v_1$, so we let $S_2 = \{v, v_1, v_2\}$), then finding v_3 that is 3rd closest to v ($v_3 \neq v_1, v_2$, so we let $S_3 = \{v, v_1, v_2, v_3\}$), $\ldots$, finding v_n that is the nth closest to v and v_n is adjacent to u giving the shortest path to u.

4. a proof by induction shows that Dijkestra's Algorithm gives a shortest path (there could be multiple paths of the same length between two vertices)

5. also, finding a circuit of shortest length that visits each vertex in a weighted graph is the famous TSP problem (traveling salesman problem) = finding a minimum Hamiltonian cycle.

6. one can use Kruskal's algorithm to find a Hamiltonian cycle that has a small weight: create a Hamiltonian cycle C by choosing the smallest edge of the graph to belong to C, then choose the next smallest edge and add it to C, $\ldots$, keep choosing the next smallest edge without closing a cycle until all vertices are incident with exactly two edges producing the cycle C (all vertices belong to the cycle C).

9.7 Planar Simple Graphs

1. Consider the problem of joining vertices without edges crossing (like placing wires on a single electronic circuit board without any wires touching except at their end vertices). That is, we want to answer the question of what graphs can be drawn in plane without any edges crossing?

2. a planar graph is a graph that can be drawn in the plane without any edges crossing (a crossing of edges is an intersection of edges at other points than the common vertex)

3. a plane graph is a particular representation of a planar graph in plane

4. note that a graph can be planar even if its regular representation is not planar. For example the planar graph $K_{2,2}$ can have a planar representation as C_4

5. any planar representation of a particular graph splits the plane into the same number of regions. Thus there is a connection between the number of edges, vertices and regions of a planar graph: Euler formula:

$$r + v - e = 2$$

6. Corollary: If G is a connected planar graph with v vertices and e edges, then

$$e \leq 3v - 6.$$

7. the converse of the above Corollary is not true. To see a counterexample, note that the graph $K_{3,3}$ is not planar, however it satisfies the relation $e \leq 3v - 6$

8. however, the contrapositive of the Corollary is very helpful: if $e > 3v - 6$, then G is nonplanar.

9. Corollary: If G is a connected planar graph, then there is a vertex u such that $\deg u \leq 5$.

10. Corollary: If G is a connected planar graph with no triangles (cycles on 3 vertices), with v vertices, and e edges, then

$$e \leq 2v - 4.$$

    (Again, the contrapositive of the corollaries are useful in proving a graph is not planar)

11. two very important nonplanar graphs: $K_{3,3}$ (see Example 3 for a proof where the graphs divides the plane into regions), and K_5 (as a consequence of the contrapositive of the above corollary)

12. Kuratowski's Theorem (characterization for planar graphs): $K_{3,3}$ and K_5 are non-planar, and so are any graphs that contain them as subgraphs or are homeomorphic to either $K_{3,3}$ or K_5.

13. two graphs G and H are homeomorphic if G can be obtained from H (or backwards) by a sequence of subdivisions. A subdivision of an edge is the replacement of the edge (a path on two vertices) with a path on three vertices, that is inserting a vertex in the middle of the edge.

9.8 Graph Colorings (for simple graphs)

1. We would like to find the minimum number of colors needed to properly color maps or other objects of interest that can be modeled by a graph. A proper coloring is an assignment of colors to the vertices of the graph, so that no two adjacent vertices use the same color. In coloring regions of a map, we use different colors for each region (the vertices of the graph) and regions that share a common boundary (the corresponding vertices are adjacent by an edge) have to be colored differently (however, if there is just a corner that they share, the two regions could use the same color).

2. another question we'll be answering in this section searches for the minimum number of slots needed to offer a particular schedule considering that people can't be part of two events in the same time (for example scheduling game matches, final exams, class schedules, or flights)

3. the chromatic number $\chi(G)$ of a graph G is the minimum number of colors needed in a proper coloring of G

4. The four color theorem: The chromatic number of a planar graph is at most four.

5. and so any map can be (properly) colored with at most 4 colors (when we say a graph is colored, we implicitly assume a proper coloring)

6. the scheduling problem can be modeled using graph colorings: vertices that represent activities that share common participants/audience are colored differently so that the activities are not scheduled to happen at the same time. The minimum number of colors needed to color the graph gives the minimum number of slots needed to schedule the activities.